Attribute Certificates APIs

Attribute Certificate Generation API

Library Functions

Functions to get information

- **`#define PEM_read_X509AC(fp,x,cb,u) (X509AC *)PEM_ASN1_read( \`**
  `(char *(*)())d2i_X509AC,PEM_STRING_X509AC,fp,(char **)x,cb,u)`

Read an attribute certificate in PEM format from a file pointer.

- **`X509AC_ISSUER_SERIAL* X509_get_basecertID(X509 *x)`**

Takes a X509 certificate and extracts the X509AC_ISSUER_SERIAL structure (or base cert ID)

- **`X509_NAME *X509AC_get_issuer_name(X509AC *a)`**

Obtain the X509_NAME of the issuer placed in a->info->issuer->d.v1Form when the attribute certificate is version 1, and from a->info->issuer->d.v2Form->issuer when the attribute certificate is version 2.

- **`X509_NAME *X509AC_get_holder_entity_name(X509AC *a)`**

Search a->info->holder->entity (stack of GENERAL_NAME) for a valid Directory Name

- **`X509AC_ISSUER_SERIAL *X509AC_get_holder_baseCertID(X509AC *a)`**

Returns a->info->holder->baseCertID structure of type X509AC_ISSUER_SERIAL.

- **`ASN1_BIT_STRING *X509AC_get_holder_objectDigestInfo(X509AC *a)`**

Returns a->info->holder->objectDigestInfo structure of type ASN1_BIT_STRING.

- **`X509AC_ISSUER_SERIAL *X509AC_get_issuer_baseCertID(X509AC *a)`**

Returns a->info->issuer->d.v2Form->baseCertID. This parameter is only available when the certificate is of version 2. For version 1 this parameter is not present.

- **`ASN1_BIT_STRING *X509AC_get_issuer_objectDigestInfo(X509AC *a)`**

Returns a->info->issuer->d.v2Form->digest. This parameter is only available when the certificate is of version 2. For version 1 this parameter is not present.

- **long X509AC_get_version(X509AC *a)**
- **int X509AC_set_version(X509AC *a, long version)**

Get and set the version of the certificate.

Functions to set information

There are three ways of providing holder information. The first one is to set the entity (GENERAL_NAME) structure with a valid directory name (X509_NAME), the second one is to set the BaseCertId structure that contains a name (X509_NAME), serial number and/or UniqueID info of the certificate that belongs to the holder the third is to set the ObjectDigestInfo.

```
ASN1_SEQUENCE(X509AC_HOLDER) = {
  ASN1_IMP_OPT(X509AC_HOLDER, baseCertID, X509AC_ISSUER_SERIAL, 0),
  ASN1_IMP_SEQUENCE_OF_OPT(X509AC_HOLDER, entity, GENERAL_NAME, 1),
  ASN1_IMP_OPT(X509AC_HOLDER, objectDigestInfo,
     X509AC_OBJECT_DIGESTINFO, 2)
} ASN1_SEQUENCE_END(X509AC_HOLDER)
```

- **int X509AC_set_holder_entity_name(X509AC* a, X509_NAME *name)**

Places a X509_NAME into a->info->holder->entity.

- **int X509AC_set_holder_serialNumber(X509AC *x, ASN1_INTEGER *serial)**

Set the serial number in a->info->holder->baseCertID->serial.

- **int X509AC_set_holder_name(X509AC* a, X509_NAME *name)**

Set the name into a->info->holder->baseCertID->issuer structure.

- **int X509AC_set_holder_objectDigestInfo(X509AC *a, X509AC_OBJECT_DIGESTINFO *odig)**

Set the object digest info of the basecertID structure for the holder of the attribute certificate.

- **int X509AC_set_holder_baseCertID(X509AC* a, X509AC_ISSUER_SERIAL *bci)**

Set the whole Base Cert ID structure.

There are two ways of providing issuer information that depends on the version of the attribute certificate. The first one is to set the v1Form (GENERAL_NAME) structure with a valid directory name (X509_NAME), the other one is to set the v2Form that can be a BaseCertId structure that contains a name (X509_NAME), serial number and/or

uniqueID info of the certificate that belongs to the holder or a X509_NAME or an objectDigestInfo.

```
ASN1_CHOICE(X509AC_ISSUER) = {
      ASN1_SEQUENCE_OF(X509AC_ISSUER, d.v1Form, GENERAL_NAME),
      ASN1_IMP(X509AC_ISSUER, d.v2Form, X509AC_V2FORM, 0)
} ASN1_CHOICE_END(X509AC_ISSUER)


ASN1_SEQUENCE(X509AC_V2FORM) = {
  ASN1_SEQUENCE_OF_OPT(X509AC_V2FORM, issuer, GENERAL_NAME),
  ASN1_IMP_OPT(X509AC_V2FORM, baseCertID, X509AC_ISSUER_SERIAL, 0),
  ASN1_IMP_OPT(X509AC_V2FORM, digest, X509AC_OBJECT_DIGESTINFO, 1)
} ASN1_SEQUENCE_END(X509AC_V2FORM)
```

- **int X509AC_set_issuer_baseCertID(X509AC* a, X509AC_ISSUER_SERIAL *bci)**

  Takes a baseCertID structure and set the issuer info of the attribute certificate.

- **int X509AC_set_issuer_name(X509AC* a, X509_NAME *name)**

  Set the name into the issuer information space. Depending on the version of the certificate it will be inserted in v1Form or in v2From->issuer.

General tools to fill up some of the necessary structures:

- **int X509AC_set_GENERAL_NAME_name(GENERAL_NAMES *gens, X509_NAME *name)**

  Introduce a X509_NAME into a GENERAL_NAMES structure.

- **int X509AC_set_baseCertID_name(X509AC_ISSUER_SERIAL *bci, X509_NAME *name)**

  Introduce a X509_NAME into a BaseCertId structure.

- **int X509AC_set_baseCertID_serial(X509AC_ISSUER_SERIAL *bci, ASN1_INTEGER *serial)**

  Introduce the serial number into a BaseCertId structure.

- **int X509AC_set_baseCertID_issuerUniqueID(X509AC_ISSUER_SERIAL *bci, ASN1_BIT_STRING *uid)**

  Introduce a unique id into a BaseCertId structure.

Attribute functions

- **X509_ATTRIBUTE * X509AC_get_attr(X509AC *a, int idx)**

  Get the X509_ATTRIBUTE that occupies the position idx in the stack.

- **int X509AC_add_attribute_by_NID(X509AC *a, int nid, int atrtype, void *value)**

Create and add an attribute based in its NID.

- **int X509AC_add_attribute(X509AC *a, X509_ATTRIBUTE *attr)**
- **int X509AC_add_X509_ATTRIBUTE(X509AC *a, X509_ATTRIBUTE *attr)**

Add an attribute to the stack in the attribute certificate.

- **ASN1_TYPE *X509AC_ATTRIBUTE_get0_type(X509_ATTRIBUTE *attr, int idx)**

Get a pointer to the ASN1_TYPE structure of the first attribute value of the attribute placed in the position idx.

- **void *X509AC_ATTRIBUTE_get0_data(X509_ATTRIBUTE *attr, int idx, int atrtype, void *data)**

Get a pointer to the data of the first attribute value of the attribute placed in the position idx.

- **int X509AC_get_attributecount(X509AC *a)**

Get the attribute count present in a attribute certificate.

Extensions:

- **int X509AC_add_extension(X509AC *a, X509_EXTENSION *ex, int loc)**

Add a X509_EXTENSION to the certificate X509_EXTENSION stack.

Signature

- **int X509AC_sign_rsa(X509AC *a, RSA *rsa, EVP_MD *md)**
- **int X509AC_sign_pkey(X509AC *a, EVP_PKEY *pkey, EVP_MD *md)**

These functions sign the attribute certificate using a RSA key or a EVP_PKEY.

Presentation

- **void X509AC_print(X509AC *ac)**

Prints to stdout the information present in a attribute certificate.

- **int GENERAL_NAMES_print(FILE *out, GENERAL_NAMES *gens)**
- **int GENERAL_NAME_print(FILE *out, GENERAL_NAME *gen)**

Other:

```
int X509AC_X509_NAME_dup(X509_NAME **xn, X509_NAME *name)
```